CLAIMS

What is claimed is:

1. A system, comprising:

a first processor having cache memory;

a second processor having cache memory and a coherence buffer that can be enabled and disabled by the first processor; and

a memory subsystem coupled to the first and second processors;

wherein for a write transaction originating from the first processor, the first processor enables the second processor's coherence buffer, and information associated with the first processor's write transaction is stored in the second processor's coherence buffer to maintain data coherency between the first and second processors.

2. The system of claim 1 wherein for write transactions originating from the second processor, the second processor sends a write exception to the first processor to maintain data coherency with an L1 cache system included in the first processor.

3. The system of claim 1 wherein the information stored in the coherence buffer is written to the second processor's cache memory.

4 . The system of claim 3 wherein, while writing the data from the second processor's coherence buffer to the cache memory, the second processor stalls if the second processor accesses the cache memory concurrently with the information being written.

5. The system of claim 1 wherein the information stored in the second processor's coherence buffer includes an address and the second processor invalidates a line in the second processor's cache corresponding to the address.

6. The system of claim 1 wherein the first processor disables the second processor's coherence buffer upon completing a write transaction to shared data.

6. The system of claim 1 wherein the first processor enables the first processor's coherence buffer upon originating a write transaction to an area of shared memory in the memory subsystem.

7. The system of claim 1 wherein the memory subsystem comprises an L2 memory subsystem and information is stored in the second processor's coherence buffer concurrently with the write transaction completing to the L2 memory subsystem.

8. The system of claim 1 wherein a control bit is associated with the coherence buffer and is programmable by the first processor to enable or disable the coherence buffer..

9. A cache coherency method usable in a multi-processor system, comprising:

when a first processor originates a write transaction to shared data, enabling a second processor's coherence buffer, and storing information associated with the first

processor's write transaction in the second processor's coherence buffer to maintain data coherency between the first and second processors; and

when the second processor originates a write transaction to shared data, sending a write exception to the first processor to cause the first processor to write data into cache local to the first processor.

10. The method of claim 9 wherein the information comprises an address associated with the write transaction originated by the first processor and the method further comprises invalidating a line in the second processor's cache corresponding to the address.

11. The method of claim 9 wherein the information comprises a data value being written by the write transaction originated by the first processor and the method further comprises writing the data value to cache memory in the second processor.

12. The method of claim 9 wherein the first processor disables the coherence buffer in the second processor after completing the write transaction originated by the first processor.